Electronic News Journal on Reasoning about Actions and Change

Editor: Erik Sandewall

Volume 1, Number 3

Contributions during October, 1997

ISSN, print: 1403 - 2031
on-line: 1403 - 204X

Linköping University Electronic Press
http://www.ep.liu.se/ej/enrac/1997/03/

Publication Methods

The Electronic News Journals (ENJ) are a medium for exchange of scientific information and debate. In particular, they serve as the forum where articles received by the ETAI (Electronic Transactions on Artificial Intelligence) are discussed publicly for review.

ENJ's are primarily published as WWW pages in HTML encoding, since they are intended for on-line use. In particular, they contain considerable numbers of links to other pages and structures on the net: articles that are available on-line, home pages of conferences and of individual researchers, links to other part of the ETAI structure, and so on. However, they also contain parts that can be read without clicking the hot links, for example, the debate contributions.

The present version of the News Journal is a derivative, formatted representation and is intended to be printed out on paper and read off-line. Due to the limitations of the paper medium, only some of the WWW links have been retained as footnotes. There are also some other differences of minor importance between the HTML version and the present one. – In order to make practical use of the WWW links, as well as to see and use other links in the structure, please retrieve the on-line ENJ from the following URL:

`http://www.ida.liu.se/ext/etai/actions/njl/`

which contains a table of back issues of ENJ's and Newsletters on Reasoning about Actions and Change.

Table of Contents

Dates of Publication

Since the date of publication may be understood either as the date of first public appearance, or as the day of reproduction on paper in many copies, and since both of these definitions may be difficult to apply in the case of electronic publication, we make the following clarifying statement.

The contents of the present issue were put on-line in their original, HTML version during the month of October, 1997. Then the contents were edited and formatted, resulting in the present, formatted version which was published on February 12, 1998, in two concurrent editions: an on-line edition and a paper edition. The on-line edition was timestamped electronically and put on-line by Linköping University Electronic Press at the URL specified on page (i). The paper edition was obtained by printing the on-line edition on a standard computer printer. It was reproduced in 200 copies, legally archived, and made available for distribution.

Vol. 3: 47-48, 31.10.1997
http://www.ida.liu.se/ext/etai/actions/njl/1997/10/

The Month of October

Summary of News Journal Contents

The month of October has been characterized, in the Newsletter for Reasoning about Actions and Change, by several submissions of articles and vivid discussions. As reflected in the present issue of the News Journal, the following is what has happened.

ETAI Received Research Articles

Two articles have been received by the ETAI area of "Reasoning about Actions and Change" during the month of October, one written by Michael Thielscher, the other by Antonis Kakas and Rob Miller. The News Journal contains the summary (longer than an abstract) for each of the articles, and the discussion protocol containing questions and answers for the latter article during this month. (There was no interaction re Thielscher's article during October).

An ETAI Received Research Note

Our ETAI area has received a research note by Paolo Liberatore, also included in this News Journal issue. Research notes are shorter than full articles and need not be self-contained to the same extent as articles often are; notes may rely on one or more background articles for the introduction to the topic, definition of notation, and even for the reference list. Since it is not reasonable to publish each research note as a separate publication, the appearance in the News Journal serves as the primary publication of a research note.

The background of Liberatore's research note are interesting as a concrete case of ETAI publication practices. This is discussed in more detail in the section on the evolution of the Electronic News Journal and Newsletter (last section of this issue).

NRAC Panel on Theory Evaluation

The NRAC workshop at this year's IJCAI conference featured several panel discussions, including one session chaired by Leora Morgenstern on the topic of Theory Evaluation: "by what criteria should theories of actions and change be evaluated?" As agreed at the workshop, the discussion in that panel continues in this electronic colloquium. The present News Journal contains three position statements by panelists at the workshop (Leora Morgenstern, David Poole, and Erik Sandewall), an additional position statement by Pat Hayes in response to the three panelists, and then a protocol of the ensuing discussion which was held in the Newsletter during October.

NRAC Panel on Ontologies for Actions and Change

The NRAC workshop also included a panel on ontologies for actions and change. Some of the contributions to the on-line discusson on theory evaluation addressed topics that more properly belong to the ontologies panel. The present editor therefore decided to open the Ontologies panel concurrently with the first one. Contributions have been received for both discussions, but the present News Journal presents them through separate protocols. For the ontologies panel, we again have three position statement, followed by a discussion protocol. In one case, I have used the editor's prerogative to divide a discussion contribution into two parts, one for each panel, in an attempt to keep each of the panels somewhat focussed.

Monthly Selected News

The "selected news of the month" is a standing headline for this News Journal, and in some of the previous issues it was a major part of the contents. However, in the present issue the selected news session is omitted because of lack of contents.

References to Articles Published Elsewhere

References to articles published elsewhere is another standing headline which has been omitted in the present issue. Your editor has been so busy with the panel debates that there was little time left for scanning journals looking for articles in our area, and of course the conference season is over.

Vol. 3: 49-50, 31.10.1997
http://www.ida.liu.se/ext/etai/received/actions/002/

Michael Thielscher:

A Theory of Dynamic Diagnosis

Summary of the article

The original version of the full article has been published by Linköping University Electronic Press, and is permanently available at
http://www.ep.liu.se/ea/cis/1997/008/

Diagnosis in general requires more than just passively observing the behavior of a faulty system. Often it is necessary to actively produce observations by performing actions. Diagnosing then amounts to reasoning about more than a single state of the system to be examined. We propose to capture this dynamic aspect by appealing to Action Theory. A formal system description consists of a *static* and a *dynamic* part. The former introduces the system components and their static relations in form of so-called state constraints, like, for instance,

$$\mathtt{active}(\mathtt{relay}_1) \equiv \mathtt{closed}(\mathtt{switch}_1)$$

stating that a particular relay is active if and only if a corresponding switch is closed. The dynamic part of a system description specifies the actions which can be used to manipulate the system's state. These definitions are accompanied by so-called action laws, which focus on the direct effects. State constraints like the above then give rise to additional, indirect effects of actions, which we accommodate according to the theory of causal relationships [Thielscher, 1997b]. E.g., this causal relationship is a consequence of our example state constraint:

$$\mathtt{closed}(\mathtt{switch}_1)\ \underline{\text{causes}}\ \mathtt{active}(\mathtt{relay}_1)$$

Informally speaking, it means that whenever $\mathtt{closed}(\mathtt{switch}_1)$ occurs as direct or indirect effect of an action, then this has the additional, indirect effect that $\mathtt{active}(\mathtt{relay}_1)$. Generally, causal relationships are successively applied subsequent to the generation of the direct effects of an action until a satisfactory successor state obtains.

In this way, the reactions of a system under healthy condition are modeled as indirect effects, so-called *ramifications*, of actions. Under abnormal circumstances—i.e., if certain aspects or components of the system are faulty—one or more of these ramifications fail to materialize. We introduce an abnormality fluent ab by which we account for such exceptions to both state constraints and the ramifications they trigger. Thus our example constraint from above, for instance, may read weaker—e.g., to the effect that

$$\neg \mathtt{ab}(\mathtt{resistor}_1) \wedge \neg \mathtt{ab}(\mathtt{relay}_1) \supset [\mathtt{active}(\mathtt{relay}_1) \equiv \mathtt{closed}(\mathtt{switch}_1)]$$

where $\mathtt{ab}(\mathtt{resistor}_1)$ and $\mathtt{ab}(\mathtt{relay}_1)$ represent an abnormal failure of a corresponding resistor and the relay itself, respectively. This weakening transfers to our expectations regarding indirect effects: The aforementioned causal relationship becomes

$$\mathtt{closed}(\mathtt{switch}_1)\ \underline{\text{causes}}\ \mathtt{active}(\mathtt{relay}_1)\ \underline{\text{if}}\ \neg \mathtt{ab}(\mathtt{resistor}_1) \wedge \neg \mathtt{ab}(\mathtt{relay}_1)$$

An important contribution of this paper, now, is a proof that due to the phenomenon of causality straightforward global minimization of abnormality—which is suitable for static diagnosis—is problematic in case of dynamic diagnosis. This raises a challenge much like the one raised by the famous Yale Shooting counter-example in the context of the Frame Problem. Meeting this challenge is inevitable when searching for 'good' diagnoses.

As a solution, we adapt from a recent causality-based solution to the Qualification Problem the key principle of *initial minimization*. All instances of the abnormality fluent are assumed false initially but may be indirectly affected by the execution of actions. In this way, our theory of dynamic diagnosis suitably exploits causal information when generating diagnoses. Our theory moreover respects available knowledge of the *a priori* likelihood of component failures. Since it is often difficult if not impossible to provide precise numerical knowledge of probabilities, we deal with qualitative rather than quantitative information, and we do not rely on complete knowledge. Such possibly incomplete information as to different degrees of abnormality is formally represented by a partial ordering among the instances of the abnormality fluent.

For the entire theory there exists a provably correct axiomatization based on the Fluent Calculus paradigm and which uses Default Logic to accommodate the nonmonotonic aspect of the diagnostic problem.

Electronic News Journal on Reasoning about Actions and Change 51
Vol. 3: *51-54*, 31.10.1997
http://www.ida.liu.se/ext/etai/received/actions/003/

Antonis Kakas and Rob Miller:

Reasoning about Actions, Narratives and Ramification

Summary of the article

The original version of the full article has been published by Linköping University Electronic Press, and is permanently available at
http://www.ep.liu.se/ea/cis/1997/010/

This paper shows how the Language $\mathcal{E}$ [Kakas and Miller, 1997] may be extended to deal with ramifications, and how domain descriptions written in this extended language may be translated into Event Calculus style logic programs. These programs are shown to be sound even when only incomplete information is given about some initial state of affairs.

The Language $\mathcal{E}$ was developed partly in response to the Language $\mathcal{A}$ [Gelfond and Lifschitz, 1993], which was introduced as the first in a family of "action description languages". Action description languages (such as $\mathcal{A}$) inherit an ontology from the Situation Calculus, whereas the Language $\mathcal{E}$ inherits its ontology (which includes an independent flow of time) from Kowalski and Sergot's Event Calculus [Kowalski and Sergot, 1986]. $\mathcal{E}$ can therefore be regarded as a basic or kernel "event description language". It was developed in the belief that the use of, and comparison between, different ontologies is important in the study of formal reasoning about actions. The semantics of $\mathcal{E}$, like that of $\mathcal{A}$, is model-theoretic, and divorced from computational considerations.

The Basic Language $\mathcal{E}$

The paper begins by reviewing the basic Language $\mathcal{E}$. $\mathcal{E}$'s vocabulary includes a set of fluent constants, a set of action constants, and a partially ordered set of time-points. Basic Language $\mathcal{E}$ domain descriptions can include three types of propositions: t-propositions ("t" for "time point"), h-propositions ("h" for "happens"), and c-propositions ("c" for "causes"). For example, the following domain description (about taking a photograph) contains 1 t-proposition, 3 h-propositions and 2 c-propositions:

```
NOT Picture holds-at 1
Load happens-at 2
Look happens-at 5
Take happens-at 8
Load initiates Loaded
Take initiates Picture when {Loaded}
```

The model theoretic semantics of $\mathcal{E}$ ensures that (for example) this domain description entails the t-proposition

```
Picture holds-at 10
```

The notions of an "initiation point" and a "termination point" are central to $\mathcal{E}$'s semantics. For example, in all models of the above domain, 2 is an initiation point for `Load` and 8 is an initiation point for `Picture`. Time can be discrete or continuous, and need not be linear. Indeed, as a special case time may be modelled as a branching structure of sequences of action constants. This allows the "simulation" in $\mathcal{E}$ of the Language $\mathcal{A}$, by writing and reasoning about t-propositions such as

```
Picture holds-at [Load, Look, Take]
```

Describing Indirect Effects in $\mathcal{E}$

The remainder of the paper discusses an extension of $\mathcal{E}$ to include a fourth type of statement called an r-proposition ("r" for "ramification"). R-propositions express permanent constraints or relationships between fluents. In formalisms which allow for such statements, the effects of actions may sometimes be propagated via groups of these constraints. This gives rise to the "ramification problem", i.e. the problem of adequately and succinctly describing these propagations of effects whilst retaining a solution to the frame problem.

R-propositions are statements of the form

```
L whenever {L1, ..., Ln}
```

The intended meaning of this statement is "at every time-point that L1, ..., Ln, L holds, and hence every action occurrence that brings about L1, ..., Ln also brings about L". Hence the semantics of $\mathcal{E}$ is extended so that from a static point of view r-propositions behave like classical constraints, but they are unidirectional in terms of the way they propagate temporal change initiated by action occurrences.

This is achieved in the main by appropriately extending the definitions of an initiation and a termination point. These definitions are now recursive, or in terms of least fixed points. R-propositions thus provide a simple, succinct method of expressing domain constraints, and the corresponding semantics behaves in a satisfactory way for a range of examples found in the literature.

The use of r-propositions is illustrated in the paper with two examples. The second of these is Thielscher's electric circuit example [Thielscher, 1997]. This example is of interest because it presents difficulties for what Thielscher describes as "categorisation-based" approaches to ramification. In the (Language $\mathcal{E}$ version of the) example, the permanent configuration and dynamic behaviour of an electric circuit is described by r- and c-propositions such as

```
Light whenever {Switch1, Switch2}
Relay whenever {Switch1, Switch2}
-Switch2 whenever {Relay}
CloseSwitch1 initiates Switch1
OpenSwitch1 terminates Switch1
CloseSwitch2 initiates Switch2 when {-Relay}
```

If `Switch2` already holds (i.e. switch number 2 is connected) and a `CloseSwitch1` action occurs, say at time-point T1, the extended semantics of $\mathcal{E}$ ensures that (in all models) the effect of this event is propagated through the first of the r-propositions above, so that `Light` becomes true. This is because the least fixed point definition of an initiation point ensures that T1 is an initiation-point for `Switch1`, and hence (by the recursive definition) an initiation point for `Light` by the first r-proposition above.

Logic Program Translations

The paper gives a translation method from $\mathcal{E}$ domain descriptions into logic programs, and gives a proof of the correctness of the translation (as regards derivation of t-propositions) for a wide class of domains. As in [Kakas and Miller, 1997], over-zealous application of logic programming's closed world assumption is avoided by representing negated fluents inside a meta-level `HoldsAt` predicate. For example, `Relay holds-at 2` is translated as `HoldsAt(Pos(Relay),2)` and `-Relay holds-at 2` is translated as `HoldsAt(Neg(Relay),2)`. C-propositions such as

```
CloseSwitch2 initiates Switch2 when {-Relay}
```

are translated into two clauses:

```
Initiates(CloseSwitch2,Switch2,t) <-
    HoldsAt(Neg(Relay),t).
```

```
PossiblyInitiates(CloseSwitch2,Switch2,t) <-
    not HoldsAt(Pos(Relay),t).
```

The first of these clauses gets used to compute changes in the truth value of `Switch2` and other fluents via occurrences of `CloseSwitch2`. The second gets used in the computation of persistence of truth values. (Similar techniques are used in a number of other logic program translations of action formalisms.)

A soundness property is proved for the logic program translations of a general class of domain descriptions, which may include r-propositions. It is stated in terms of SLDNF-derivability: if there is a finite SLDNF derivation of `HoldsAt(Pos(F),T)` (respectively `HoldsAt(Neg(F),T)`) from the program, then `F holds-at T` (respectively `-F holds-at T`) is entailed from the original domain description.

For the examples given in the paper, these logic programs are more-or-less directly executable in Prolog. The relevant Prolog listings are available at
http://www.dcs.qmw.ac.uk/~rsm/abstract15.html

Vol. 1: *55-59*, 31.10.1997
http://www.ida.liu.se/ext/etai/received/actions/003/

Protocol of on-line discussion during October, 1997 about the following research article:

Antonis Kakas and Rob Miller:

Reasoning about Actions, Narratives and Ramification

Q1. **Michael Thielscher** (24.10)

Antonis and Rob,

I have a question concerning the notion of initiation and termination points in case ramifications are involved. If my understanding of your Definition 14 is correct, then there seems to be a problem with undesired mutual justification. Take, as an example, the two r-propositions

```
dead  whenever  -alive
-alive  whenever  dead
```

Suppose there are no other propositions, in particular no events, then

```
H(0)= {alive, -dead}
H(1)= {-alive, dead}
```

seems to satisfy all conditions for being a model. The two uncaused changes justify each other: 0 is an initiation point for *dead* since 0 is a termination point for *alive*, and vice versa.

Finding some least fixpoint, which you mention after the definition, seems therefore vital for the correctness of the definition itself. However, the corresponding operator must not have an interpretation as argument. So I would think that instead of defining the notions of "initiation and termination points for F in H relative to D" one should define "initiation and termination points for F relative to D," that is, without reference to some H.

A1. **The authors** (30.10)

Hello Michael, Thanks for your comments about Definition 14 of initiation and termination points. You are of course right to say that the definition requires the least fixed point construction, so perhaps we should have made this explicit within the definition itself. We omitted this from the paper in an attempt not to overload the definition with too much formalism, but perhaps its omission is causing more rather than less confusion. (Hudson Turner emailed us a similar comment to your's a little while ago.)

So yes, the initiation and termination points are defined by a least fixed point construction (along the lines we say after the definition). The version of the definition that makes this explicit is specified in detail in the research note on page 66-67 of this News Journal issue. You'll see that the operator corresponding to the least fixed point does indeed have an interpretation as argument. But there's no problem with this, because the interpretation is already fixed at the beginning of the definition. It's necessary include this argument in order to deal with preconditions of c-propositions. For example, consider the following domain (with time as the naturals):

```
Take initiates Picture when {Loaded}
Take happens-at 2
-Picture holds-at 1
```

We want 2 models, one in which Loaded is true at 1, and one in which Loaded is false at 1. In the former model, 2 should be an initiation point for Picture, but in the latter it shouldn't.

Q2. **Tom Costello** (28.10)

In your paper you have three types of proposition, h, t and c-propositions. In your definition of an interpretation, you give enough information to establish truth conditions for t-propositions. The following is the obvious truth condition for t-propositions.

A t-proposition, `F holds-at T`, is true in an interpretation E, if `E(F,T) = true`.

However, you do not seem to have enough information to give truth conditions for h or c-propositions.

Consider the domain language with one time-point 0 and one fluent F and one action A. Then the domain description,

```
A happens at 0
F holds at 0
```

has one model, $(F,0) \mapsto$ true

The domain description

```
F holds at 0
```

has the same model. However, these two descriptions differ on the h propositions. Thus from an interpretation you cannot determine the set of true h-propositions.

For a logic to model distinct sets of propositions by the same structure is problematic for many reasons.

As a general point, $\mathcal{A}$ type languages are not sufficiently formal in defining when a proposition is true in a model. This has led to errors like the above in $\mathcal{A}$-type languages. Some papers have used a function from sequences of actions to sets of fluents, rather than a labeled transition function/relation from sets of fluents to sets of fluents, to give semantics to action languages. The former collapses domain descriptions that differ on causal propositions, while the latter does not. Giunchiglia, Kartha and Lifschitz are an example of the use of the latter. I know of no paper that explicitly gives truth conditions for all propositions in an $\mathcal{A}$-type language

A2. **The authors** (30.10)

Hello Tom, Thanks for your comments and observations.

Regarding your specific comments about the Language $\mathcal{E}$, then you're right - from a formal point of view there is no concept of truth or falsity as regards h- and c-propositions. So, from the definitions, it doesn't even make sense to talk about "the set of true h-propositions". For your example, the semantics simply "disregards" the h-proposition "A happens-at 0", because the occurrence of A at 0 that this represents at the syntactic level has no effects.

There's no problem with this from a formal point of view, but it does mean that $\mathcal{E}$, and languages like it, are very restrictive. That's why they're perhaps best regarded as stepping-stones towards formalisations or axiomatisations written in fuller, general-purpose logics. (However, and as we hope we and others have illustrated, they do have a use in discussing and illustrating approaches to particular issues - in our case, to ramifications - in a relatively intuitive and uncluttered way, and also in proving properties of classes of logic programs.) This is where work such as that of Kartha (translating A into various versions of the Situation Calculus) is valuable. In the case of the Language $\mathcal{A}$, Kartha's translations bring out the fact that there is an implicit completion of causal information ($\mathcal{A}$'s e-propositions) in $\mathcal{A}$'s semantics. Much the same thing is true of h- and c-propositions in $\mathcal{E}$. (This is why adding truth functions for h- and c-propositions in $\mathcal{E}$ models would be trivial but rather superfluous).

We discussed this in more detail in our first paper on $\mathcal{E}$ (in the Journal of Logic Programming paper). As we've said in both papers, it's our intention to explore these issues further by developing translations analogous to Kartha's for $\mathcal{E}$. You might also be interested to look at the papers by Kristof Van Belleghem, Marc Deneker, and Daniele Theseider Dupre, who have developed a language $\mathcal{ER}$ similar in many respects to $\mathcal{E}$, but more expressive and with a correspondingly more complex semantics (which includes truth conditions for the equivalent of h- and c-propositions). (We've described this briefly in Section 5 of our paper.)

As regards your general point about "$\mathcal{A}$ type languages", it would be interesting to get some comments from "$\mathcal{A}$ type people" about this. Perhaps "not sufficiently expressive" is a better phrase than "not sufficiently formal". (On this general theme, Mikhail Soutchanski made another good point in the recent ENRAC when he pointed out that it's much easier to combine theories of action written in classical logic with other commonsense theories, e.g. of space or shape, than if specialised logics are used.)

Q3. **Tom Costello** (30.10)

A question on the choice of approach: Why didn't you write everything in classical logic? Personally, I find it much more natural to consider classical logical languages than $\mathcal{A}$-type languages. My separakte note (pages 60-65 of this News Journal issue) is a translation of the proposed $\mathcal{E}$ language to a classical language, which I feel makes much clearer the advantages and disadvantages of the proposal.

A3. **The authors** (30.10)

Hello Tom, – We've no objection to using classical logic. Indeed, in both our $\mathcal{E}$ papers we've mentioned our intention to translate $\mathcal{E}$ into classical logic and other general-purpose formalisms, in order to gain the obvious benefits. (An obvious candidate as a target for this translation is something like the classical logic Event Calculus in [Miller & Shanahan 1996].) As you indicate in your question, different researchers will find different approaches more natural. We chose to initially express our ideas on ramification in this form because we found it relatively intuitive and uncluttered, and convenient for proving properties of logic programs that we want to use for various applications. As we've stated in our answer to your previous question and in our first paper on $\mathcal{E}$, these specialised languages are perhaps best regarded as stepping-stones towards formalisations or axiomatisations written in fuller, general-purpose logics. It's great that you have in fact used $\mathcal{E}$ in exactly this way. Please publish!

One point about your relations "init" and "term" in your classical logic translation. You say that you should take the "smallest relations ... that satisfy the above [axioms partially defining the relations]". But it turns out that this "smallest relation" idea is still not quite sufficient for eliminating the kind of anomalous models that Michael Thielscher was drawing attention to. So you really do need a least fixed point notion or equivalent somewhere in your axiomatisation, where the associated operator generates the least fixed point starting from a pair of empty sets (see our answer to Michael's question).

Of course, another reason for using the specialised language approach was to illustrate that the Language $\mathcal{A}$ type methodology could be applied using ontologies other than that of the Situation Calculus. We're not sure if authors of Language $\mathcal{A}$ type papers would reply to your question in the same way, so it would be interesting to get some other responses from this community.

Rob and Tony

Electronic News Journal on Reasoning about Actions and Change 60
Vol. 1: *60-65*, 31.10.1997
http://www.ida.liu.se/ext/etai/rac/notes/1997/12/

Putting $\mathcal{E}$ into classical logic

Tom Costello
Stanford University, CA, USA

Kakas and Miller suggest an action language, $\mathcal{E}$, which they extend to deal with ramifications. We show how their language can naturally be embedded in a classical language. Action languages have become popular, because they are claimed to be *easily readable, understandable and intuitive*. Some people (especially the author) find it more natural to write in classical logic. Thus this note transcribes the notation of Kakas and Miller to classical second order logic.

We first present a family of second order languages, and then show that this family captures (the finite part of) the $\mathcal{E}$ family.

Thus, rather than use a new class of languages, we use second order logic, with three non-empty sorts, actions a, fluents f, and times t, using g for sets of fluents, with predicates, $t \leq t'$, $holds(f,t)$, $happens(a,t)$, $initiates(a,f,g)$, and $terminates(a,f,g)$. We also add the predicate $whenever(f,g)$ later. We have a function not on fluents, which is of period two, and interacts with the $holds$ predicate [1] on fluents in the natural way.

$$\begin{array}{l} \forall f.not(not(f)) = f \\ \forall f,f'.not(f) = not(f') \rightarrow f = f' \\ \neg(F = not(F')), \;\; \text{for constants } F, F' \\ \forall f,t.holds(f,t) \equiv \neg holds(not(f),t) \end{array} \tag{1}$$

We postulate partial order axioms for $\leq$,

$$\begin{array}{l} \forall t.t \leq t' \\ \forall t,t',t''.t \leq t' \wedge t' \leq t'' \rightarrow t \leq t'' \\ \forall t,t'.t \leq t' \wedge t' \leq t \rightarrow t = t' \end{array} \tag{2}$$

[1] Note that we do not have the intuitive

$$\forall f,t,g.initiates(a,f,g) \equiv terminates(a,not(f),g),$$

as this would change the requirements on domain descriptions being consistent. That is, the domain description,

A **terminates** F_1 **when** F
A **initiates** F_1 **when** F

has models.

We define predicates $term(f,t)$ and $init(f,t)$, which captures the notion of t being an initiation point/ termination point for f.

$$\forall f,t[init(f,t) \equiv \exists a,g[happens(a,t) \wedge initiates(a,f,g) \wedge \forall f'[g(f') \rightarrow holds(f',t)]]] \tag{3}$$

$$\forall f,t[term(f,t) \equiv \exists a,g.happens(a,t) \wedge terminates(a,f,g) \wedge \forall f'[g(f') \rightarrow holds(f',t)]]] \tag{4}$$

We now write the three conditions about termination/initiation points.

$$\forall f,t_1,t_3[\forall t_2[t_1 \leq t_2 \wedge t_2 \leq t_3 \rightarrow \neg(term(f,t_2) \vee init(f,t_2))] \rightarrow holds(f,t_1) \equiv holds(f,t_3)] \tag{5}$$

$$\forall t_1,t_3,f[init(f,t_1) \wedge \forall t_2[t_1 \leq t_2 \wedge t_2 \leq t_3 \rightarrow \neg term(f,t_2)] \rightarrow holds(f,t_3)] \tag{6}$$

$$\forall t_1,t_3,f[init(f,t_1) \wedge \forall t_2[t_1 \leq t_2 \wedge t_2 \leq t_3 \rightarrow \neg init(f,t_2)] \rightarrow \neg holds(f,t_3)] \tag{7}$$

We also have unique names axioms for all our constants. We have domain closure axioms stating that every object is a constant, save for fluents, where we have an axiom that states that every fluent is a constant or the result of applying *not* to a fluent constant. We have axioms relating or negating the relation of every pair of time-points in the partial order $\leq$.

We now need to define the translation of a domain description in $\mathcal{E}$ into our second order language. To do this we need some definitions from Kakas and Miller, namely definitions 1, 2, 3, 4, 5, and 12.

We define the translation of a domain language [2] $\langle \Pi, \preceq, \Delta, \Phi \rangle$ into a set of sentences in second order logic. Our time-point constants are the objects in Π, our relation $t \leq t'$ is true when $t \preceq t'$. Our fluent constants are Φ, union the image of Φ under *not*. Our action constants are the set Δ. Our axioms are the unique names and domain closure axioms, the axiom 1, and the sentences defining $\leq$ for every pair of time-points. We denote the translation of a domain language $\mathcal{D}$ as $\mathcal{D}^*$.

Lemma: 1 *The theory of the sub-language of the translation of a domain language $\langle \Pi, \preceq, \Delta, \Phi \rangle$, whose only predicates are the equalities for each sort, and $\leq$, and whose functions are not and whose constants are all constants, is complete.*

[2] Note that if Δ, Φ or Π are infinite, then we may not be able to write down domain closure axioms. Therefore we limit ourselves to the case where they are all finite. We can easily move to non-finite domains by introducing well-orders of each domain, which would allow us to state domain closure using induction on the appropriate well-order.

Proof: Domain closure axioms give us quantifier elimination, all ground terms are decided by unique names axioms, save for fluents of the form $not(f)$, and $\leq$. Ground cases of $\leq$ are decided by assumption. This leaves equalities involving $not(f)$, which can be reduced to equalities of the form $f = not(f')$ for constant f, f', using $not(not(f)) = f$, and $not(f) = not(f') \rightarrow f = f'$. These cases are decided by 1. ∎

We define the translation of an h-proposition A **happens-at** T, as $happens(A, T)$. This is a well formed second order logic formula, in our language, as all time-points and actions in the domain language are constants in our second order language.

We define the translation of a t-proposition L **holds-at** T as $holds(L, T)$ if L is a constant, and as $\neg holds(F, T)$ if $L = \neg F$.

We define the translation of a c-proposition A **initiates** F **when** C, as

$$initiates\left(A, F, \lambda f. \bigvee_{F' \in C} f = F'\right).$$

We define the translation of a c-proposition A **terminates** F **when** C, as

$$terminates\left(A, F, \lambda f. \bigvee_{F' \in C} f = F'\right).$$

Finally we define the translation of an r-proposition L **whenever** C as,

$$whenever\left(F, \lambda f. \bigvee_{F' \in C} f = F'\right)$$

if L is a fluent constant F, and as $whenever\,(not(F), \lambda f. \bigvee_{F \in C} f = F)$ if L is $\neg F$ for some fluent constant F.

We define the translation of a domain description D, as the set of translations of its elements, and we denote it D^*. Given a set of propositions P, we define their translation P^*, by applying the above translation.

We use the usual notation $Circ(A; P)$ for the minimization of P in the theory A.

Lemma: 2 *A domain description, D, stated in $\mathcal{D}$, contains a h-proposition, A* **happens-at** *T if and only if $Circ(P^*; happens) \models happens(A, T)$, where P is the set of h-propositions in D.*

Proof:

The circumscription is equivalent to,

$$\forall a, t. happens(a, t) \equiv \bigvee_{A \textbf{ happens-at } T \in P} a = A \wedge t = T$$

The lemma follows immediately. ∎.

Lemma: 3 *A domain description, D, stated in $\mathcal{D}$ contains a c-proposition,*
A **initiates** F **when** C
if and only if

$$Circ(P^*; initiates) \models initiates\left(A, F, \lambda f. \bigvee_{F \in C} f = F\right)$$

where P is the set of c-propositions in D.

Proof:

The circumscription is equivalent to

$$\forall a, f, g[initiates(a, f, g) \equiv \bigvee_{\alpha} a = A \wedge f = F \wedge \forall f'[g(f') \equiv \bigvee_{F \in C} f' = F]]$$

where α stands for

A **initiates** F **when** $C \in P$

The other formulas agree with the original, and, of course, I am The lemma follows immediately. ∎.

The analogous lemma also holds for *terminates*.

This gives the notion of a model of $\mathcal{E}$ without ramifications. We use Germanic font for second order structures, and use $holds^{\mathcal{M}}$ for the interpretation of a predicate *holds* in the structure $\mathcal{M}$.

Theorem: 1 *Let D be a domain description, stated in $\mathcal{D}$. Then its models are isomorphic to the interpretation of holds in models of $\mathcal{D}^*$, 1, 3 , 4, 5, 6, 7, D^*, $Circ(P^*; happens)$, $Circ(Q^*; initiates)$, $Circ(Q^*; terminates)$, where P and Q are the h, c propositions in D respectively.*

Proof: We consider the conditions on being a model of D, and show that they agree with the second order conditions.

Firstly, by lemma 1 the sorts, under the interpretation of $=$ and $\leq$, give a domain language, as by completeness there is a unique model that is exactly three sets and a partial ordering.

Secondly, *holds* is of the correct type to be an interpretation, namely a function from Φ, Δ to $\{\text{true}, \text{false}\}$. Thus, we need only check the four conditions on being a model. However, these depend on the notion of an initiation/termination point. We first show that $init(F,T)$ ($term(F,T)$) is true in a model $\mathcal{M}$, exactly then T is an initiation(termination) point for F relative to the interpretation $holds^{\mathcal{M}}$.

We only consider initiation points as the argument for termination points is exactly analogous.

The definition of an initiation point is in terms of whether there is an h-proposition of the form A **happens-at** T, and a c-proposition of the form A **initiates** F **when** C. However, by Lemma 2 and Lemma 3, this is exactly equivalent to the truth of whether our circumscription implies $happens(A,T)$, and $initiates\,(A,F,\lambda f.\bigvee_{F\in C} f = F)$. There are no other sentences that mention *happens* or *initiates* save for the circumscription, and D^*. However, if D^* is consistent, then D^* implies $happens(A,T)$ only if the circumcription implies it. Therefore, if the sentences, $\mathcal{D}^*$, 1, 3, 4, 5, 6, 7 are consistent, then the entire theory entails, $happens(A,T)$ if and only if there is an h-proposition of the form A **happens-at** T. The sentences $\mathcal{D}^*$, 1 are immediately consistent, as the domain language provides a model. 4 and 3 can be taken to be definitions, and are thus their addition does not make the other sentences $\mathcal{D}^*$ and 1 inconsistent. Thus we need to show that 5 6 and 7 are consistent with $\mathcal{D}^*$, 1 and 4 and 3. If D has an interpretation, this serves as a model. Thus we may take the sentences to be consistent. Thus we have shown that the entire theory entails, $happens(A,T)$ if and only if there is an h-proposition of the form A **happens-at** T. The case of *initiates* is similar.

Thus, we are reduced to checking that the interpretation satisfies C at T. However, this is equivalent to $\forall f.\bigvee_{F\in C} f = F \rightarrow holds(f,T)$, and thus the existence of an initiation point is equivalent to whether $\exists a,g.happens(a,T)\wedge initiates(a,F,g)\wedge(\forall f'.g(f')\rightarrow holds(f',T))$ is true. This is the definition of $init(F,T)$, and thus $init(F,T)$ is true in $\mathcal{M}$ if and only if T is an initiation point for F in $holds^{\mathcal{M}}$.

Now we can check our four conditions. The first is immediately 5, as $T \preceq T'$ when $T \leq T'$, *holds* is defined to be equal to the interpretation, and by the above *init* and *term* are true when there are initiation and termination points. The second and third are exactly 6 and 7, by the same reasoning.

Thus we are left with the fourth condition, which follows as the interpretation is defined in terms of *holds*.

Thus, the conditions for an interpretation H to be a model of a domain description D stated in $\mathcal{D}$ are exactly the conditions for a set of fluent, time-points pairs to be the interpretation of *holds*, in models that satisfy $\mathcal{D}^*$, 1, 3, 4, 5, 6, 7, $Circ(P^*;happens)$, $Circ(Q^*;initiates)$, $Circ(Q^*;terminates)$, D^* where P and Q are the h, c propositions in D respectively. ∎

To get ramifications we need to add the predicate *whenever*. We also add an axiom,

$$\forall f,g,t.whenever(f,g)\rightarrow((\forall f'.g(f')\rightarrow holds(f',t))\rightarrow holds(f,t).$$

We also need to change the definitions of *term* and *init*.

We have a choice here, we can use the following definition, where *term* is defined as any predicate satisfying,

$$\begin{aligned}
&\forall f,t[term(f,t) \equiv \\
&\exists a,g[terminates(a,f,g) \wedge \forall f'[g(f') \rightarrow holds(f',t)]] \vee \\
&\exists C_1,C_2[whenever(not(f),\lambda f'.C_1(f') \vee C_2(f')) \wedge \\
&\quad \forall f'[C_1(f') \rightarrow holds(f',t)] \wedge \\
&\quad \forall f'[C_2(f') \rightarrow term(f',t)]]]
\end{aligned} \tag{8}$$

and *init* is defined as any predicate satisfying,

$$\begin{aligned}
&\forall f,t[init(f,t) \equiv \\
&\exists a,g[initiates(a,f,g) \wedge \forall f'[g(f') \rightarrow holds(f',t)]] \vee \\
&\exists C_1,C_2[whenever(not(f),\lambda f'.C_1(f') \vee C_2(f')) \wedge \\
&\quad \forall f'[C_1(f') \rightarrow holds(f',t)] \wedge \\
&\quad \forall f'[C_2(f') \rightarrow init(f',t)]]]
\end{aligned} \tag{9}$$

This suffers from the counter-example suggested by Thielscher. However, my reading of the paper suggests that we take the smallest relations *term* and *init* that satisfy the above, as the definition, as the authors suggest that the least fixed-point be taken.

The proof that this captures the same notions as the definition of a model in $\mathcal{E}$ is essentially the same as before, save that we have some more conditions to check.

Electronic News Journal on Reasoning about Actions and Change 66
Vol. 1: *66-67*, 31.10.1997
http://www.ida.liu.se/ext/etai/rac/notes/1997/15/

Answer to a Question by Tom Costello

Antonis Kakas[1] *and Rob Miller*[2]
[1] *University of Cyprus;* [2] *Queen Mary and Westfield College, London, England*

Hello Michael,

Thanks for your comments about Definition 14 of initiation and termination points. You are of course right to say that the definition requires the least fixed point construction, so perhaps we should have made this explicit within the definition itself. We omitted this from the paper in an attempt not to overload the definition with too much formalism, but perhaps its omission is causing more rather than less confusion. (Hudson Turner emailed us a similar comment to your's a little while ago.)

So yes, the initiation and termination points are defined by a least fixed point construction (along the lines we say after the definition). Here's a version of the definition that makes this explicit:

Definition 14 [Initiation/termination points] Let H be an interpretation of $\mathcal{E} = \langle \Pi, \preceq, \Delta, \Phi \rangle$, and $D = \langle \gamma, \eta, \tau, \rho \rangle$ be a domain description. Let $\mathcal{W}$ be the set $2^{\Phi\times\Pi} \times 2^{\Phi\times\Pi}$ and let the operator $\mathcal{F} : \mathcal{W} \mapsto \mathcal{W}$ be defined as follows. For each, $\langle \mathcal{I}n, \mathcal{T}e \rangle \in \mathcal{W}$ denote $\mathcal{F}(\langle \mathcal{I}n, \mathcal{T}e \rangle)$ by $\langle \mathcal{I}n', \mathcal{T}e' \rangle$. Then for any $F \in \Phi$ and $T \in \Pi$, (F, T) is in $\mathcal{I}n'$ (respectively in $\mathcal{T}e'$) iff one of the following two conditions holds.

1. There is an $A \in \Delta$ such that (i) there is both an h-proposition in η of the form "A **happens-at** T" and a c-proposition in γ of the form "A **initiates** F **when** C" (respectively "A **terminates** F **when** C") and (ii) H satisfies C at T.

2. There is an r-proposition in ρ of the form "F **whenever** C" (respectively "$\neg F$ **whenever** C") and a partition $\{C_1, C_2\}$ of C such that (i) C_1 is non-empty, for each fluent constant $F' \in C_1$, $(F', T) \in \mathcal{I}n$, and for each fluent literal $\neg F' \in C_1$, $(F', T) \in \mathcal{T}e$, and (ii) there is some $T_2 \in \Pi$, $T \prec T_2$, such that for all T_1, $T \preceq T_1 \preceq T_2$, H satisfies C_2 at T_1.

Let $\langle \mathcal{I}n^f, \mathcal{T}e^f \rangle$ be the least fixed point of the operator $\mathcal{F}$ starting from the empty tuple $\langle \emptyset, \emptyset \rangle$. T is an *initiation-point* (respectively *termination-point) for F in H relative to D* iff $(F, T) \in \mathcal{I}n^f$ (respectively $(F, T) \in \mathcal{T}e^f$).

□

So the operator corresponding to the least fixed point does indeed have an interpretation as argument. But there's no problem with this, because the interpretation is already fixed at the beginning of the definition. It's necessary include this argument in order to deal with preconditions of c-propositions. For example, consider the following domain (with time as the naturals):

$$\begin{gathered} Take \textbf{ initiates } Picture \textbf{ when } \{Loaded\} \\ Take \textbf{ happens-at } 2 \\ \neg Picture \textbf{ holds-at } 1 \end{gathered}$$

We want 2 models, one in which *Loaded* is true at 1, and one in which *Loaded* is false at 1. In the former model, 2 should be an initiation-point for *Picture*, but in the latter it shouldn't.

Rob and Tony.